TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 10/10 pp750–762
DOI: 10.26599/TST.2018.9010144
Volume 24, Number 6, December 2019

A Deep Adaptive Learning Method for Rolling Bearing Fault Diagnosis Using Immunity

Yuling Tian* and Xiangyu Liu

Abstract: The extraction of rolling bearing fault features using traditional diagnostic methods is not sufficiently comprehensive and the features are often chosen subjectively and depend on human experience. In this paper, an improved deep convolutional process is used to extract a set of features adaptively. The hidden multi-layer feature of deep convolutional neural networks is also exploited to improve the extraction features. A deterministic detection of low-confidence samples is performed to ensure the reliability of the recognition results and to decrease the rate of false positives by evaluating the diagnosis of the deep convolutional neural network. To improve the efficiency of the continuous learning elements of the rolling bearing fault diagnosis, a clone learning strategy based on cloning and mutation operations is proposed. The experimental results show that the proposed deep convolutional neural network model can extract multiple rolling bearing fault features, improve classification and detection accuracy by reducing the false positive rate when diagnosing rolling bearing faults, and accelerate learning efficiency when using low-confidence rolling bearing fault samples.

Key words: deep learning; fault diagnosis; feature extraction; clone selection strategy

1 Introduction

Large electromechanical devices are complicated nonlinear dynamic systems characterized by uncertainty, nonlinearity, and time-variance, resulting in complicated fault statuses with many interfering factors. Rolling bearings are important parts of mechanical equipment and often present as the source of faults[1]. An intelligent fault diagnosis system can predict faults to a certain extent, and provide an early warning in the event of a weak fault. In so doing, it can effectively avoid any chain reaction that might otherwise arise from the fault, and obviate the need for replacement of the entire piece of equipment.

With the development of intelligent technology, many algorithms based on deep learning have been applied to fault diagnosis[2]. The advantage of deep learning methods lies in the integration of feature extraction and classification, and the adaptive extraction of fault features from fault signals[3]. The main deep learning algorithm models with applications in the field of fault diagnosis include self-encoding, deep belief networks, Convolutional Neural Networks (CNN)[4], and long short-term memory networks[5]. Tran et al.[6] proposed a hybrid deep belief network, adopting the deep belief network for pre-training and classifing faults by the Simplified Fuzzy ARTMAP (SFAM) to recognize the single and combined faults of suction and discharge valves in a reciprocating compressor. A method based on stacked denoising auto-encoding and the Gath-Geva clustering algorithm for roller bearing fault diagnosis without a data label has also been proposed[7]. Meng et al.[8] suggested obtaining the training data using a new data pre-processing method, changing the unit numbers of each layer to alter the hyperparameter of autoencoders, and adopting multivariate norm penalties

• Yuling Tian and Xiangyu Liu are with the College of Information and Computer, Taiyuan University of Technology, Taiyuan 030000, China. E-mail: tianyuling@tyut.edu.cn; 18713340041@163.com.

∗ To whom correspondence should be addressed.

Manuscript received: 2018-11-27; revised: 2019-01-09; accepted: 2019-01-20

to get a better sparse representation, and finally reusing the data points between the adjacent samples to improve the fault identifying rate. A fault classification method using the compressed sensing and a deep neural network has been presented in Ref. [9]. By utilizing the deep belief network's learning ability, the proposed method can adaptively fuse multi-feature data and identify various bearing faults[10]. Tran et al.[11] presented an approach implementing vibration, pressure, and current signals for fault diagnosis of the valves in reciprocating compressors. A pre-processing step has been proposed to improve the performance of the CNN by extracting Envelope Spectrums (ES) on the raw fault signals. As ES demodulates the signals to provide the fault information, the CNN can learn to extract distinctive features to diagnose bearing defects effectively[12]. When using CNN for fault diagnosis, the two-Dimensional (2D) images are generated by the time and frequency domain waveform of the raw fault signals and used as the input data of CNN to extract the fault features[13, 14]. A two-layered scheme for the bearing faults identification has also been proposed in Ref. [15], combining the feature pool and the sparse stacking automatic encoder to extract more discriminating information from the raw vibration signals and identify the multiple faults. When an unknown fault occurs, the deep learning model is unable to adjust and the model must be retrained, resulting in poor model adaptation. Inspired by the biological immune system, the artificial immune system provides new approaches for intelligent fault detection and diagnosis[16]. Such approaches, providing noise tolerance, successive learning, memory acquisition, and requiring no non-self samples[17], have the advantage of an evolutionary learning mechanism and the potential to provide novel methods to solve fault diagnosis problems[18].

In this paper, to solve the problem of the time needed for the detection and learning of an adaptive algorithm model, the combination of a Deep Convolutional Neural Network (DCNN) and antibody immunity is proposed for the real-time rolling bearing fault diagnosis. The DCNN is used to extract the characteristics of the time domain and frequency domain signals of rolling bearings, realizing a direct mapping from raw data to diagnostic result. In the detection phase, the fault types are determined by comparing the time domain and frequency domain diagnosis results. In the learning phase, the efficiency of learning unknown faults is improved by a cloning strategy and a continuous mutation operation.

2 Self-Adaptive DCNN Detection Model

The Self-adaptive DCNN (Sel-DCNN) fault diagnosis model is made up of three main parts. The first part is feature extraction and recording, mainly using the constructed time and frequency domain DCNN models to perform waveform signal feature extraction and record the extracted features and diagnostic results. The second part is the diagnosis of known faults, which includes two functions: firstly, to evaluate the records and determine the fault type; secondly, to store the fault detectors obtained through learning. Each fault type corresponds to an antibody population and each antibody population has a tag corresponding to the fault type, which facilitates quick matching when a known fault occurs. The third part is the learning of unknown faults. Its main function is to complete the learning of unknown faults and generate detectors. In the process of antibody production, an antigen-centered antibody initialization generation strategy is suggested with the main purpose of accelerating the optimization of antibodies during the learning phase[19]. In the process of fault diagnosis, the concept of an evaluation threshold based on multiple diagnosis results is also adopted to avoid the occurrence of misjudgments arising from mutations in the operating status. Figure 1 shows the overall architecture of the model. First, it uses the DCNN time domain and frequency domain model for diagnosis, and records the extracted features and preliminary diagnostic results. In the fault diagnosis part, it diagnoses the fault types based on the preliminary diagnostic results and records of the previous stage. Known fault detection and unknown fault learning are both realized in the model. The unknown fault detection uses the detectors in the memory knowledge base. After an unrecognized situation occurs, fault learning is performed. The main operations include cloning, mutation, and selection. Unknown faults are learnt to produce a mature detector and place it in the memory knowledge base for rapid detection when this type of fault occurs again.

2.1 Parameter selection for the DCNN fault diagnosis model

When using a DCNN, the network's structure is built first, then the parameters are chosen for the specific application field. During the process of constructing

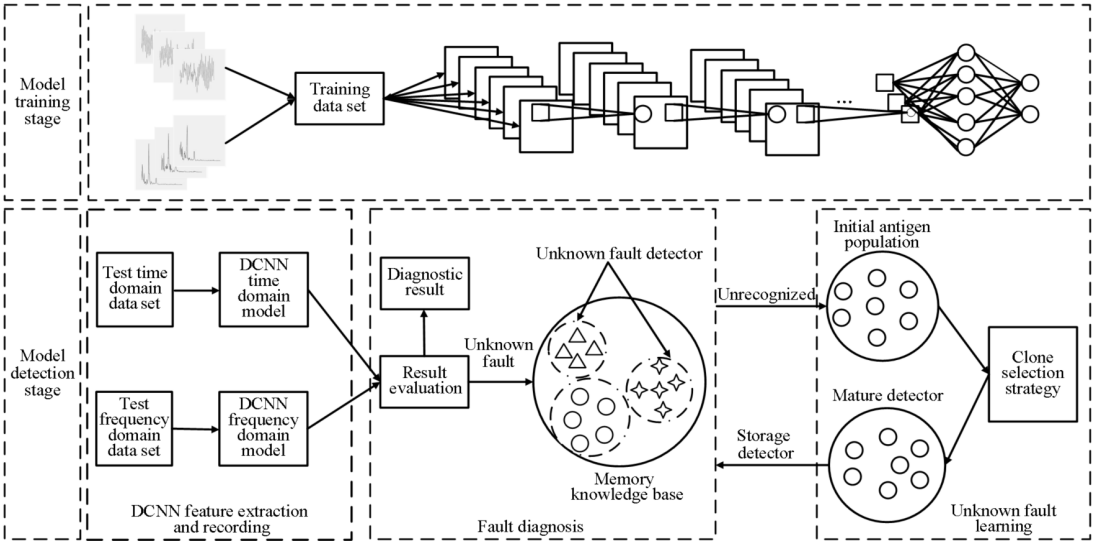

Fig. 1 Self-adaptive DCNN fault diagnosis model.

fault diagnosis models based on DCNNs, parameter selection is based on a 32×32 frequency domain signal. The parameters consist of the number of feature maps of the convolutional layer and the size of the convolutional kernel. The number of convolution layers is 4 and the selection of the number of feature maps is made in accordance with three different models. The first model type is an "incremental" model, in which the number of feature maps of each convolutional layer is gradually increasing; the number of feature graphs extracted is then 30×50×80×100. The second "constant" model means that the number of feature maps of each convolution layer is fixed; the number of selected feature maps in this model is 60. The third "decreasing" model is one in which the number of feature maps of each convolutional layer is gradually decreasing, in this case, the extracted number is 100×80×50×30. The convolution operation is mainly used for the feature extraction of images. The convolution kernel size is usually selected after determining the number of convolution layer feature maps. In this case, there are three convolution kernels of different sizes (i.e., 3×3, 4×4, and 5×5). The pooling kernel is set to the most commonly used size, which is 2×2 in this case. Along with the increasing of iteration times, the appropriate number of training batches is increasing. The selections of the subsequent time domain model and frequency domain model are completed based on the selections of these three parameters. Table 1 shows the accuracy of each parameter set for the training and test set. The appropriate parameters are selected based on the test results.

According to the accuracy of the training set and the test set, which can be read from Table 1, it can be seen that when the feature map is selected as "decreasing" (the model with better overall performance), the accuracy is higher than that when either of the other two models are selected. DCNNs return better diagnostic results when the convolution kernel is 4×4. When selecting the number of training batches, the detection

Table 1 Main parameters selection of network model.

	Training batches	Convolution layer number	Number of feature maps	Convolution kernel size	Training set accuracy (%)	Test set accuracy (%)
Number selection of feature maps	50	4	30×50×80×100	3×3	99.87	98.86
	50	4	60×60×60×60	3×3	99.77	98.14
	50	4	100×80×50×30	3×3	99.83	99.19
Selection of convolution kernel size	50	4	100×80×50×30	3×3	99.83	99.19
	50	3	100×80×50	4×4	99.93	99.13
	50	3	100×80×50	5×5	99.96	98.62
Selection of training batch size	40	3	100×80×50	4×4	99.94	98.76
	60	3	100×80×50	4×4	99.96	99.16
	80	3	100×80×50	4×4	99.96	98.75

accuracy has reached its optimum when the number of batches increases or is equal to 60. Beyond this, even if the number of batches is increased further, there is no significant impact on the test results.

2.2 Size selection of the input time domain and frequency domain signal

After selecting the main parameters of the DCNN, the next step is to determine the number of layers and the size of the input time domain and frequency domain signals. DCNN does not have a unified model at the time of construction, and there is no standard definition for the size of an input image. In different application areas, the DCNN operates with different network structures and different input image sizes are chosen in accordance with the specific situation. The network structure is determined by contrasting the detection accuracy of training set and test set with different input image sizes. According to the results of the parameter selection, the optimal size of the convolution kernel is 4×4, the size of the pooling kernel is the commonly used 2×2, the number of feature maps suggests a "decreasing" model, and the optimal number of training batch sets is 60. Five different input signal sizes and corresponding network layer numbers are set. Tables 2 and 3 report the diagnostic results of the time and frequency domain on the training set and test set, respectively. The time domain signal size selection is shown in Table 2, while the frequency domain signal size selection is shown in Table 3.

It can be seen from Tables 2 and 3 that different input time domain and frequency domain waveforms correspond to different network model structures. In the time domain waveform diagnosis results, the use of the same network structure for different waveform sizes is mainly achieved through a complementary edge operation. As the input waveform signal increases, the network layer is deepened, and a nine-layer network

Table 2 Time domain signal size selection.

Image size (pixel)	Network layer numbers	Convolution kernel size	Pooling size	Train set accuracy (%)	Test set accuracy (%)
15×15	7	4×4	2×2	98.83	96.10
20×20	8	4×4	2×2	99.69	98.19
25×25	8	4×4	2×2	99.87	97.95
28×28	9	4×4	2×2	99.88	98.00
32×32	9	4×4	2×2	99.83	98.19

Table 3 Frequency domain signal size selection.

Image size (pixel)	Network layer numbers	Convolution kernel size	Pooling size	Train set accuracy (%)	Test set accuracy (%)
15×15	7	4×4	2×2	99.80	97.90
20×20	8	4×4	2×2	99.93	99.10
25×25	8	4×4	2×2	99.91	99.38
28×28	9	4×4	2×2	99.93	99.14
32×32	9	4×4	2×2	99.96	99.16

model structure with a size of 32×32 achieves the highest accuracy of recognition of the time domain signal. When choosing the frequency domain network model, there is almost no difference in the accuracy of the training set, but a slightly larger difference can be seen in the accuracy of the test set. The highest accuracy is achieved when the input frequency domain waveform size is 25×25 and the network structure has eight layers. The recognition effect of the frequency domain diagnosis model is better compared to that of the time domain diagnosis model.

2.3 Design of the time domain and frequency domain fault diagnosis models

The time domain diagnosis model has a nine-layer structure, as shown in Fig. 2. The first layer is the input layer, and the input signal of the real-time domain is a grayscale image, the size of which is 32×32. The second layer is a convolutional layer. After one convolution it obtains 100 feature maps of

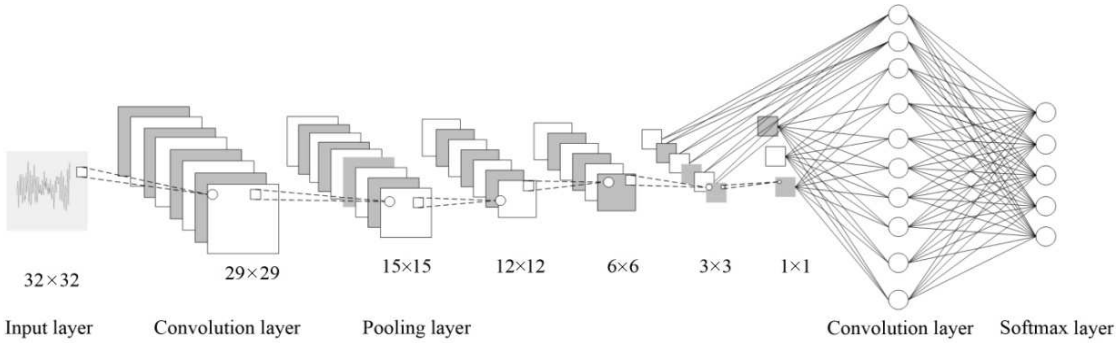

Fig. 2 DCNN time domain model.

29×29 size. The third layer is the pooling operation. Because the pooling kernel is 2×2, the dimension of the previous convolution layer cannot be divisible, and a complementary edge operation is performed with 100 feature maps of 15×15 size obtained by the pooling operation. The fourth layer is a convolution operation and obtains 50 feature maps of 12×12 size. The fifth layer gets 50 feature maps of 6×6 size after the pooling operation; after the sixth layer convolution, 30 feature maps of 3×3 size are obtained. The seventh layer gets a one-dimensional vector by complementing edge and convolution operations. The eighth layer of the neural unit consists of two parts; one part is obtained using a 1×1 convolution kernel for the seventh layer, and the other part is obtained by the convolution operation for the sixth layer feature maps. The ninth layer is the Softmax layer, and each output unit corresponds to a fault type. In order to obtain fuller features, a full convolutional layer instead of a fully connected layer is used in the network's construction. Through the convolution operation, the common features of the sixth and seventh layers are integrated into the Softmax function for classification, thus improving the recognition accuracy.

Tables 4 and 5 detail the construction of the time domain and the frequency domain signal model. The input waveform signal used in the frequency domain diagnostic model has a size of 25×25, for a total of eight layers. The convolution and pooling processes in the frequency domain and the time domain are the same so that the details will not be repeated here.

2.4 DCNN model initialization and training process

The DCNN model begins training once construction is completed. The data set uses the processed time domain and frequency domain waveform signal data. In this paper, we select data set from the Western Reserve University Bearing Data Center, the initialization and training processes are as follows.

Table 4 Time domain signal model structure.

Layer number	Number of feature maps	Feature map size	Layer number	Number of feature maps	Feature map size
1	1	32×32	6	30	3×3
2	100	29×29	7	30	1×1
3	100	15×15	8	50	1
4	50	12×12	9	7	1
5	50	6×6			

Table 5 Frequency domain signal model structure.

Layer number	Number of feature maps	Feature map size	Layer number	Number of feature maps	Feature map size
1	1	25×25	5	50	4×4
2	100	22×22	6	30	1×1
3	100	11×11	7	50	1
4	50	8×8	8	7	1

- After the data set is divided and the grayscale is preprocessed, the obtained sample is stored under the folder.
- Deep convolutional neural network model is constructed according to the selected parameters and the network structure.
- Data set is divided into a training set and a test set, using a 4:1 ratio.
- Steps for convolution and pooling operations are set to 1 and 2, respectively, and the number of offsets is set to equal the number of feature maps.
- Convolution kernel performs random initialization, the bias initialization is set to zero, the learning rate is set to 0.001, and other related parameters are set.
- In the forward propagation process, the convolution operation uses the following formula:

$$x_j^i = f(\sum_{i \subset M_j} x_i^{l-1} \times K_{ij}^l + b_j^l) \tag{1}$$

where l represents the layer number, K is the convolution kernel, M_j is the receptive field of the input layer, b represents the offset; x indicates a feature map, and $f(\cdot)$ represents the activation function.

The pooling operation uses the largest pooling method. The selection of the activation function is ReLU, which avoids the problem of the disappearing gradient and has a faster convergence speed. Its calculation formula is Eq. (2), with x representing the input value of the neuron,

$$f(x) = \max(0, x) \tag{2}$$

- The DCNN model is constructed by pre-processed time domain and frequency domain, using the random gradient descent training method to adjust the error through back propagation. Finally, the trained rolling bearing diagnostic network model is obtained.
- The test set is tested by the trained deep convolutional neural network model, thus the influence of a deep convolutional neural network on fault diagnosis and feature extraction of rolling bearings is verified.

3 Immune Deterministic Detection

3.1 Problem definition

In the adaptive DCNN fault diagnosis model, the

input data is waveform signal of the time domain and frequency domain. The feature extraction of the time domain and frequency domain waveform signals is performed using the DCNN, and the Softmax function is used for classification. When encountering an unknown fault, it is necessary to perform immune component mapping on the extracted features. In the fault diagnosis module, there is a memory knowledge base, which stores learned fault detectors, with each type of fault recorded with an antibody population and a DCNN diagnosis result. The function of the unknown fault learning module is to train the optimal antibody population, which includes the operations of cloning, mutation, and selection of antibodies.

Definition 1: Antigen. The antigen is composed of features extracted by the DCNN in the time and frequency domain. The tag of the antigen is composed of the initial diagnostic results of the feature extraction and recording module, represented by two characteristic bits. The features are derived from the input data of the Softmax layer. The feature numbers extracted by the time domain and frequency domain recognition models of the DCNN are the number of training data classifications. The antigen Ag is defined by the following expression:

$$Ag = (TD, FD, \boldsymbol{T}_1, \ldots, \boldsymbol{T}_n, \boldsymbol{Ft}_1, \ldots, \boldsymbol{Ft}_n, R_{ag}) \quad (3)$$

where TD and FD are the results of the time domain and frequency domain recognition of the DCNN, respectively. $\boldsymbol{T}_n$ and $\boldsymbol{Ft}_n$ are the feature vectors extracted by DCNN in the time domain and the frequency domain, respectively. And mapped as the characteristic attributes of the antigen, R_{ag} is the radius of the antigen.

Definition 2: Antigen population. When unknown faults occur, the waveforms will appear as continuous abnormal situations, and the features extracted and the diagnostic results will be recorded each time. Each record is an antigen, and there are N stored records that are directly mapped by an antigen population. The Agp is the antigen population characterized by

$$Agp = (TD_r, FD_r, Ag_1, Ag_2, \ldots, Ag_n) \quad (4)$$

where TD_r and FD_r are the diagnosis results of the DCNN in the time domain and the frequency domain, respectively. The values of TD_r and FD_r are the most frequently fault type of N diagnostic records, and then the antigen records are stored. Each antigen Ag_n corresponds to one sample.

Definition 3: Antibody population. In fault diagnosis, the antibody corresponds to a fault detector. The resulting antibodies are preferably able to cover all antigens and have a high affinity. We propose an improved initial antibody production strategy in which the gravity center of the triangle consisted of three antigens becomes the center of a new antibody based on the antigen population, i.e., radius. The antibody population is Abp and defined as

$$Abp = \frac{\mathbf{C}^3_{Ag_i} + \mathbf{C}^3_{Ag_j} + \mathbf{C}^3_{Ag_k}}{3} \quad (5)$$

Definition 4: Average affinity. In the artificial immune algorithm, the affinity is used to indicate the degree of match between the antibody and the antigen. However, the affinity in fault diagnosis is the degree of match between the detector and the extracted feature. Among these, the antibody and the antigen have the same vector dimensions, while the average affinity has a higher reliability and can avoid the misjudgment caused by detection and recognition of the single antibody edge. The training phase is the average affinity of the antibody for all antigens, but the detection phase is the average affinity of the antigen for all antibodies. The average affinity uses a method based on the distance defined by the antibody and the antigen,

$$F_i = \frac{n}{\sum_{j=1}^{n} |Ab_j - Ag_i|} \quad (6)$$

where n represents the size of the initial antibody population, Ab_j represents an antibody in the antibody population, Ag_i represents an antigen in the population, and F_i represents the average affinity of an antigen.

Definition 5: Clone operation. After calculating the average affinity of the initial antibody, the average affinity is ranked. According to the defined affinity calculation formula, the larger the affinity, the more the antibodies and antigens match. The size of an antibody clone is proportional to the average affinity, K is a clonal factor, Num is the initial antibody population size, and AbN_i is the initial population size of the antibody,

$$AbN_i = K \times Num \times F_i \quad (7)$$

Definition 6: Mutation operation. The mutation probability is defined based on the average affinity. The greater the average affinity, the smaller the probability of mutation. The formula for calculating the mutation probability is

$$P_i = \frac{\alpha}{F_i} \quad (8)$$

where P_i is the probability of mutation of the antibody

i; the greater it's value, the greater the probability that the antibody will mutate. α is a mutation factor. When performing the mutation, the mutation center of the antibody is randomly selected and the mutation operation formula is as follows,

$$At(t+1)_d = At(t)_d + \varepsilon \frac{R_{ag}}{F_i} \tag{9}$$

where $At(t+1)_d$ denotes the position of the dimension d of the antibody in the $(t+1)$-th generation, $At(t)_d$ denotes the position of the d dimension of the antibody in the t-th generation, and ε is a random number between 0 and 1.

3.2 Antibody learning

When an unknown fault occurs, if the memory knowledge base cannot cognize it, immune learning of antibodies is performed to generate a new fault detector. The antibody learning process is as follows.

(1) Map the features extracted from the rolling bearing deep convolutional neural network to the initial antigen population (*Agp*). The number of initial antigens is the recording threshold.

(2) Generate the initial antibody population using an antigen center-based antibody production strategy after the initial antigen production.

(3) Calculate the average affinity of each antibody for all antigens according to the cloning procedure definition. In this step, the corresponding antibodies will be cloned according to the cloning operation. The number of cloned antibodies is proportional to the average affinity between the antibody and the antigen; the higher the average affinity, the larger the number of clones of the antibody.

(4) Set a random probability threshold for each cloned antibody. When the calculated mutation probability is greater than the corresponding random threshold, perform the mutation for the cloned antibody, otherwise there is no mutation. The mutation probability is inversely proportional to the average affinity calculation value. Having set the threshold, recalculate the average affinity of each antibody, both cloned and antigen.

(5) Compute the affinity for the newly generated antibodies and sequence in descending order. Select the highest affinity (L) antibody as the next generation of the initial antibody population.

(6) Repeat the cloning and mutation operation. When the number of iterations reaches the specified number or the antibody affinity satisfies the specified threshold, terminate the execution.

(7) A certain number of new detectors will have been generated by the completion of the algorithm. Add these new detectors to the memory knowledge base.

3.3 Adaptive DCNN detection

With the record value threshold set to N and the evaluation threshold to D, the adaptive DCNN detection approach consists of the following main steps.

(1) Perform the recognition using the trained time domain and frequency domain models of the DCNN, record the diagnostic results and extracted features each time. The record set is defined as $Rec = \{Re_1, Re_2, \ldots, Re_n\}$, and Re_n is defined in the form of antigens.

(2) When the number of recordings reaches N, evaluate the diagnostic results of the records and compare the tag of each Re_n in the set. Set the number of times as M when the diagnostic results of time domain waveform and frequency domain waveform are consistent. The time domain model diagnoses that the maximum number of fault types is $T_{\max}$, the frequency domain is $F_{\max}$, and the category tag corresponding to the maximum number of times is the fault base, represented by two characteristic bits.

(3) If M is less than the evaluation threshold D, then an unknown fault has occurred, so subject the record value to antigen mapping and perform step (5).

(4) If M is greater than the evaluation threshold D and the evaluation result is in a normal state, clear the record and perform the detection for the following period. If the evaluation result is a fault type, define the current state as the fault type and clear the record.

(5) When there is an unknown fault type, check whether $T_{\max}$ and $F_{\max}$ are both greater than the evaluation threshold D. If so, check the fault base against the tag bit of the antibody population in the memory knowledge base. If they match, then perform the antibody detection steps; otherwise, perform the unknown fault learning procedure. If $T_{\max}$ and $F_{\max}$ are not both greater than the evaluation threshold D, call the untagged antibody population detection. If it can be identified, output the diagnosis result and clear the record. If it is still not recognized, perform the online learning procedure.

(6) If the memory knowledge base can identify an unknown fault, consider the diagnosis to be completed and clear the record; otherwise, perform the online fault learning procedure.

(7) If the memory knowledge base cannot identify the unknown fault type, the feature data extracted by the deep convolution neural network is passed to the unknown fault learning module to generate a mature detector.

4 Experimental Design and Analysis

4.1 Data pre-processing

The experimental data set is the rolling bearing data set provided by Case Western Reserve University Data Center[20]. The raw data is the vibration acceleration signal, which is composed of hashed data points. When generating the input layer data of the fault diagnosis model, a time domain waveform is taken at every interval and subjected to a Fourier transform to obtain a corresponding frequency domain waveform. The fault types in the data set include normal state, inner ring fault, outer ring fault, and rolling element fault. According to the diameter of the fault space region, we classify each fault type into one of three states: slight fault (0.1778 mm), moderate fault (0.3556 mm), and severe fault (0.5334 mm). The data used for training are in the normal, slight fault and severe fault states. We select 1500 samples for each type of data, and 10 500 samples in time domain and frequency, respectively. The size of signal map is 224×224 generated in each time domain and frequency domain. In training the DCNN, the time domain and frequency domain signals are resized as input data to the DCNN. The waveforms of the time domain and frequency domain of the bearing faults are listed in Figs. 3 and 4; in the data set the fault diameter is 0.1778 mm. It can be seen from Figs. 3 – 10 that the differences in the time domain waveforms are mainly manifested in the amplitude of the vibration, whereas in the frequency domain they appear in different frequency bands. From the overall view of the signal waveforms in the time domain and the frequency domain, it can be seen that the various faults appear very differently based on the fault type, which provides a basis for the feasibility of using DCNNs for fault diagnosis and of the proposed algorithm model.

4.2 DCNN feature extraction validation

4.2.1 Comparison of DCNN and feature extraction techniques

The results of the experiment comparing the deep convolutional neural network method with the commonly used feature extraction fault diagnosis techniques are shown in Table 6. Table 6 lists the experimental results of the time domain recognition

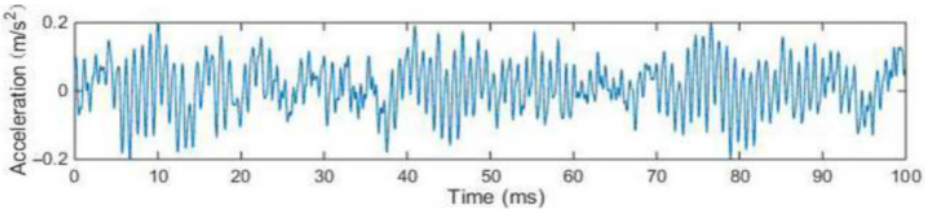

Fig. 3 Normal state time signal.

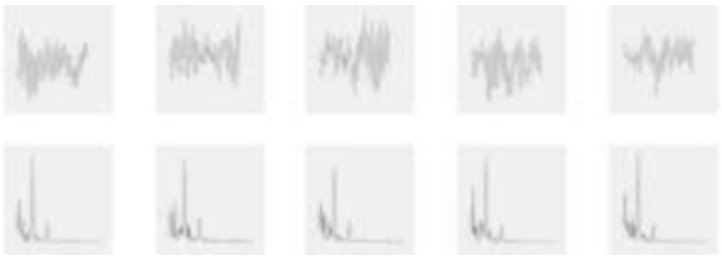

Fig. 4 Normal state time and frequency domain signal sample.

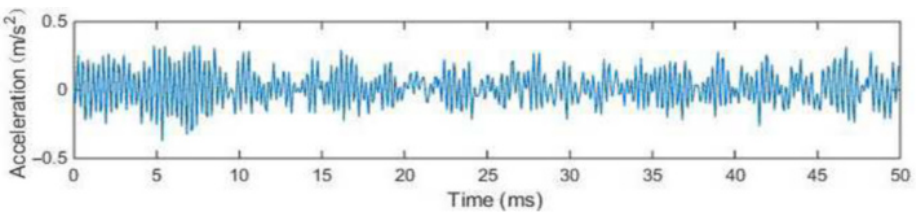

Fig. 5 Slight rolling element fault time domain signal.

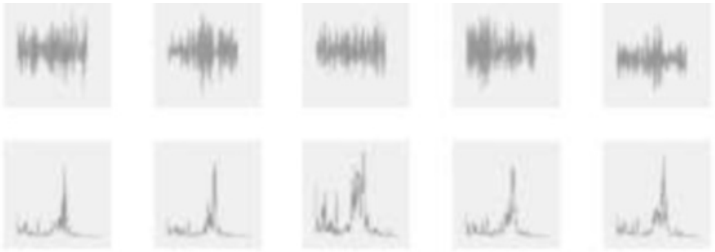

Fig. 6 Slight rolling element fault time and frequency domain signal sample.

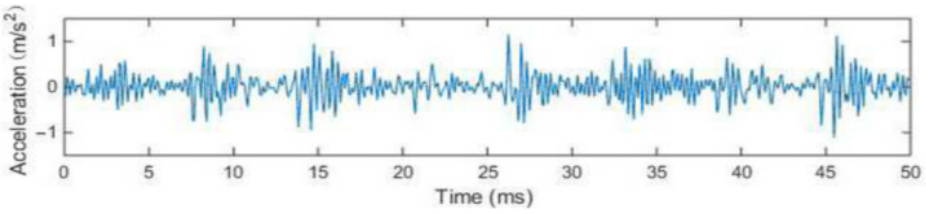

Fig. 7 Slight inner ring fault time domain signal.

Fig. 8 Slight inner ring fault time and frequency domain signal sample.

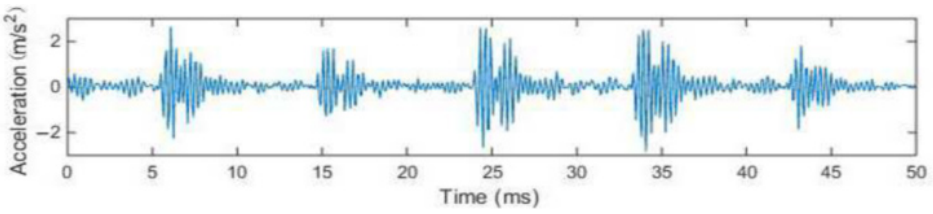

Fig. 9 Slight outer ring fault time domain signal.

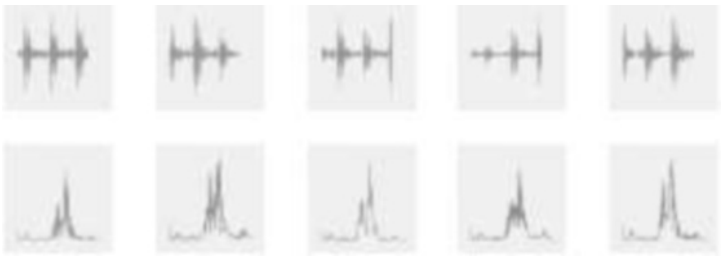

Fig. 10 Slight outer ring fault time and frequency domain signal sample.

model of DCNN (DCNN-TD) and the frequency domain recognition model of DCNN (DCNN-FD). Feature Abstraction (FA) is based on the dimensionless data feature extraction of earlier waveforms; the extracted feature indices include waveform indicators, peak indicators, pulse indicators, margin indicators, and kurtosis indicators. The features extracted by Empirical Mode Decomposition (EMD) and Ensemble Empirical Mode Decomposition (EEMD) are the energy characteristics of the decomposed IMF components. Due to the randomness of the initial parameter settings, the detection accuracy is taken as the average of ten runs. From the accuracy comparison reported in Table 6, it can be seen

Table 6 Comparison of DCNN and feature extraction techniques.

Technical method	Train set accuracy (%)	Test set accuracy (%)
DCNN-TD-Softmax	99.83	98.19
DCNN-FD-Softmax	99.91	99.38
EMD-SVM	92.25	92.00
EMD-BP	86.03	81.64
EEMD-SVM	91.56	90.67
EEMD-BP	85.19	80.86
FA-SVM	81.83	81.08
FA-BP	80.55	80.23

that the recognition rate of fault diagnosis based on the DCNN is clearly higher than both that of a BP neural network approach based on feature extraction and that of a fault feature recognition approach based on Support Vector Machines (SVM). This reflects the more comprehensive and appropriate nature of the features extracted by the DCNN, and the suitability of these extracted features for classification. Through the comparison between the DCNN model and other feature extraction technologies, it can be seen that the DCNN model has great advantages in performing fault feature extraction of rolling bearings.

4.2.2 Comparison of DCNN and Deep Belief Network (DBN)

The DBN is a model based on the probability of energy generation, comprising of multiple layers of restricted Boltzmann machines. It is also commonly used for fault feature mining and intelligent diagnosis. There are two ways to process input layer data for a deep belief network: the first is to take N most primitive data points as an input sample, represented by an N-dimensional vector; the second is to take the obtained time domain and frequency domain signal waveforms as the raw input data, and with each sample process the 2D data to generate a one-dimensional data vector. The experimental data set is shown in Table 7. Based on the DCNN model, seven types of fault data are used. When the DBN is used for waveform signal identification, the bearing fault data is a slight fault data set. Reference [21] used the DBN for bearing fault diagnosis by setting the input vectors of different sample lengths, which showed the validity of the original vibration data feature extraction. From the accuracy results of the experiment on the training and test sets, it can be seen that the CNN and DBN are both effective in identifying bearing faults to a certain extent, however, when time domain and frequency domain signals are used as input data to a deep belief network, it is not effective at classifying faults. These include time domain based on deep belief network (named DBN-TD) and frequency domain based on deep belief network (named DBN-FD).

4.3 Analysis and comparison of experimental results

4.3.1 Distinguishing the level of fault

In the case of the fault level discrimination test, the algorithm model used is based on a well-selected time domain and frequency domain DCNN model. The data used is of the types slight fault (0.1778 mm) and severe fault (0.5334 mm). The accuracy results of the experiment are shown in Tables 8 and 9. From these

Table 7 Comparison of DCNN and DBN.

Algorithm model	Input data	Network layer number	Train set accuracy (%)	Test set accuracy (%)
DCNN-TD	32×32	9	99.83	98.19
DCNN-FD	25×25	8	99.91	99.38
DBN[21]	1024	4	99.40	none
DBN-TD	20×20	4	25.00	25.00
	25×25	4	25.00	25.00
	30×30	4	25.00	25.00
DBN-FD	20×20	4	25.00	25.00
	25×25	4	25.00	25.00
	30×30	4	25.00	25.00

Table 8 Train set accuracy for various fault types.

Algorithm model	Inner ring fault (%)	Outer ring fault (%)	Rolling element fault (%)	Severe inner ring fault (%)	Severe outer ring fault (%)	Severe rolling element fault (%)
DCNN-TD	99.92	99.83	100	99.75	99.67	99.58
DCNN-FD	99.83	99.92	99.93	99.92	99.90	99.86

Table 9 Test set accuracy of various fault types.

Algorithm model	Inner ring fault (%)	Outer ring fault (%)	Rolling element fault (%)	Severe inner ring fault (%)	Severe outer ring fault (%)	Severe rolling element fault (%)
DCNN-TD	98.33	97.33	99.11	97.00	97.33	99.27
DCNN-FD	99.24	99.43	99.56	99.25	96.00	99.67

experimental results, it can be seen that the constructed DCNN can distinguish the same fault at different fault levels, indicating the ability of DCNNs to extract nuanced features.

4.3.2 Diagnosis results of the adaptive DCNN

In Table 10, the experimental data was calculated using the average of multiple runs. It can be seen from the experimental results that the recognition rates of each of the separate time domain and frequency domain fault diagnosis models reach a level above 99%. The proposed adaptive DCNN model can achieve good results in the learning of unknown faults on the premise of ensuring the accuracy of known fault recognition. It overcomes the inability of the DCNN model to identify unknown faults, and also has a lower false negative and false positive rate.

4.3.3 Comparison of experimental results

An analysis of the experimental results was performed to compare the proposed model with other methods in the literature. The first three items in Fig. 11 are the experimental results from our work, with the accuracy rate being the results from the test set. The EHHT-CNNs algorithm model uses the modified Hilbert-Huang transform to decompose the instantaneous frequency spectrum at different scales to obtain the instantaneous frequency vector, and then transforms the vector into a 2D matrix suitable for the convolutional neural network[22]. WDCNN proposed a one-dimensional deep convolutional network model and added a batch normalization layer to improve the accuracy of detection[23]. Figure 11 shows a comparison of the recognition accuracy of each of these algorithms. It can be seen from the experimental results that the recognition accuracy rate of our proposed model is higher than that of the deep convolution network alone, and also improves on the EHHT-CNNs and WDCNN algorithm models.

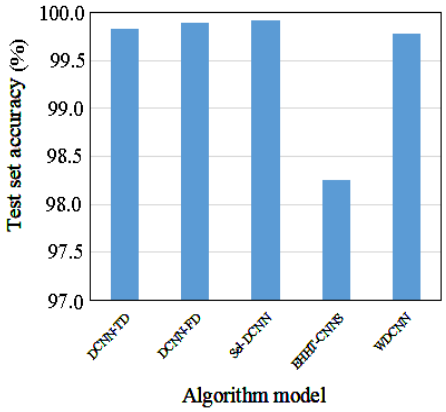

Fig. 11 Comparison of accuracy of identification.

5 Conclusion

This paper combines a deep convolutional network with an immunity algorithm for fault diagnosis by integrating signal features in the time domain and frequency domain. Based on an improved DCNN, an immune adaptive DCNN model is proposed. The new model aims to effectively identify unknown faults on the premise of ensuring the recognition of known faults. Compared with traditional feature extraction and classifier recognition in other models based on deep convolutional networks, the feature extraction process in the proposed model becomes more intelligent without relying on human factors, while also extracting more complete results and achieving a higher recognition accuracy rate. Additionally, in the fault diagnosis phase, in order to achieve a result in real time, a faster and more efficient DCNN diagnosis model is proposed by introducing a cloning and mutation strategy. Through the experimental comparison of the detection efficiency of several algorithm models, the proposed model is not only shown to be effective in classifying faults, but also has higher recognition efficiency.

Table 10 Sel-DCNN model diagnosis results.

Fault type		Train set (%)	Test set (%)	Test false positive rate (%)	Test missed rate (%)
Sel-DCNN	Known fault	99.92	99.42	0.28	0.57
	Unknown fault	none	98.32	2.16	1.85
DCNN-TD	Known fault	99.83	98.19	2.67	1.89
DCNN-FD	Known fault	99.91	99.38	0.34	0.72

Acknowledgment

This work was supported by the National Natural Science Foundation of China (No. 61472271).

References

[1] H. Long, C. Li, and H. Liu, Feature extraction method of rolling bearing fault signal based on EEMD and cloud model characteristic entropy, *Entropy*, vol. 17, no. 12, pp. 6683–6697, 2015.

[2] T. Ince, S. Kiranyaz, L. Eren, M. Askar, and M. Gabbouj, Real-time motor fault detection by 1-D convolutional neural networks, *IEEE Transactions on Industrial Electronics*, vol. 63, no. 11, pp. 7067–7075, 2016.

[3] D. Cabrera, F. Sancho, C. Li, M. Cerrada, R. V. Sanchez, F. Pacheco, and J. V. de Oliveira, Automatic feature extraction of time-series applied to fault severity assessment of helical gearbox in stationary and non-stationary speed operation, *Applied Soft Computing*, vol. 58, pp. 53–64, 2017.

[4] D. Dey, B. Chatterjee, S. Dalai, S. Munshi, and S. Chakravorti, A deep learning framework using convolution neural network for classification of impulse fault patterns in transformers with increased accuracy, *IEEE Transactions on Dielectrics and Electrical Insulation*, vol. 24, no. 6, pp. 3894–3897, 2017.

[5] D. Park, S. Kim, Y. An, and J. Y. Jung, LiReD: A light-weight real-time fault detection system for edge computing using LSTM recurrent neural networks, *Sensors*, vol. 18, no. 7, pp. 1–15, 2018.

[6] V. T. Tran, F. AlThobiani, T. Tinga, and A. Ball, Single and combined fault diagnosis of reciprocating compressor valves using a hybrid deep belief network, *Journal of Mechanical Engineering Science*, vol. 232, no. 20, pp. 3767–3780, 2018.

[7] F. Xu, W. T. Tse, and Y. L. Tse, Roller bearing fault diagnosis using stacked denoising autoencoder in deep learning and Gath-Geva clustering algorithm without principal component analysis and data label, *Applied Soft Computing*, vol. 73, pp. 898–913, 2018.

[8] Z. Meng, X. Y. Zhan, J. Li, and Z. Z. Pan, An enhancement denoising autoencoder for rolling bearing fault diagnosis, *Measurement*, vol. 130, pp. 448–454, 2018.

[9] H. O. A. Ahmed, M. L. D. Wong, and A. K. Nandi, Intelligent condition monitoring method for bearing faults from highly compressed measurements using sparse over-complete features, *Mechanical Systems and Signal Processing*, vol. 99, pp. 459–477, 2018.

[10] J. Tao, Y. Liu, and D. Yang, Bearing fault diagnosis based on deep belief network and multisensor information fusion, *Shock and Vibration*, vol. 2016, no. 7, pp. 1–9, 2016.

[11] V. T. Tran, F. Althobiani, and A. Ball, An approach to fault diagnosis of reciprocating compressor valves using Teager-Kaiser energy operator and deep belief networks, *Expert Systems with Applications*, vol. 41, no. 9, pp. 4113–4122, 2014.

[12] D. K. Appana, W. Ahmad, and J. M. Kim, Speed invariant bearing fault characterization using convolutional neural networks, in *Proc. 11th International Workshop on Multi-disciplinary Trends in Artificial Intelligence*, Gadong, Brunei, 2017, pp. 189–198.

[13] V. David, A. S. Ferrada, E. L. Droguett, and V. Meruane, Deep learning enabled fault diagnosis using time-frequency image analysis of rolling element bearings, *Shock and Vibration*, vol. 2017, pp. 1–17, 2017.

[14] M. J. Hasan and J. M. Kim, Bearing fault diagnosis under variable rotational speeds using stockwell transform-based vibration imaging and transfer learning, *Applied Sciences-Basel*, vol. 8, no. 12, pp. 1–15, 2018.

[15] S. Muhammad, K. Cheol-Hong, and K. Jong-Myon, A hybrid feature model and deep-learning-based bearing fault diagnosis, *Sensors*, vol. 17, no. 12, pp. 1–16, 2017.

[16] E. Hart and D. Davoudani, An engineering-informed modelling approach to AIS, in *Proc. 10th International Conference on Artificial Immune Systems*, Cambridge, UK, 2017, pp. 240–253.

[17] L. Montechiesi, M. Cocconcelli, and R. Rubini, Artificial immune system via euclidean distance minimization for anomaly detection in bearings, *Mechanical Systems and Signal Processing*, vols. 76&77, pp. 380–393, 2016.

[18] O. Vatefipour, A novel electric load consumption prediction and feature selection model based on modified clonal selection algorithm, *Journal of Intelligent and Fuzzy Systems*, vol. 34, no. 4, pp. 2261–2272, 2018.

[19] N. Bayar, S. Darmoul, S. H. Gabouj, and H. Pierreval, Fault detection, diagnosis and recovery using artificial immune systems: A review, *Engineering Applications of Artificial Intelligence*, vol. 46, pp. 43–57, 2015.

[20] W. A. Smith and R. B. Randall, Rolling element bearing diagnostics using the Case Western Reserve University data: A benchmark study, *Mechanical Systems and Signal Processing*, vol. 64, no. 12, pp. 100–131, 2015.

[21] W. H. Li, W. P. Shan, and X. Q. Zeng, Bearing fault identification based on deep belief network, *Journal of Vibration Engineering*, vol. 29, no. 2, pp. 340–347, 2016.

[22] Q. Li, Y. Liu, and H. J. Liang, A new fault diagnosis method based on HHT-CNNs and its application in rolling bearing fault diagnosis, in *Proceedings of the 36th Chinese Control Conference*, Dalian, China, 2017, pp. 7021–7026.

[23] W. Zhang, G. Peng, C. Li, Y. H. Chen, and Z. J. Zhang, A new deep learning model for fault diagnosis with good anti-noise and domain adaptation ability on raw vibration signals, *Sensors*, vol. 17, no. 2, pp. 425–446, 2017.

Yuling Tian is a professor of Information and Computer Department, Taiyuan University of Technology, China. She received the PhD degree from Taiyuan University of Technology, China, in 2009. She has been responsible for and participated in a number of national and provincial fund projects, and published more than 30 papers in the academic journals and conferences. Her research interests include artificial intelligent and fault diagnosis.

Xiangyu Liu is a PhD candidate of Information and Computer Department, Taiyuan University of Technology, China. His research interests include deep learning and fault diagnosis.